Power Plastics: A Hybrid Lagrangian/Eulerian Solver for Mesoscale Inelastic Flows

ZIYIN QU, University of Pennsylvania, USA and University of California Los Angeles, USA
MINCHEN LI, University of California Los Angeles, USA and Carnegie Mellon University, USA
YIN YANG, University of Utah, USA
CHENFANFU JIANG, University of California Los Angeles, USA
FERNANDO DE GOES, Pixar Animation Studios, USA

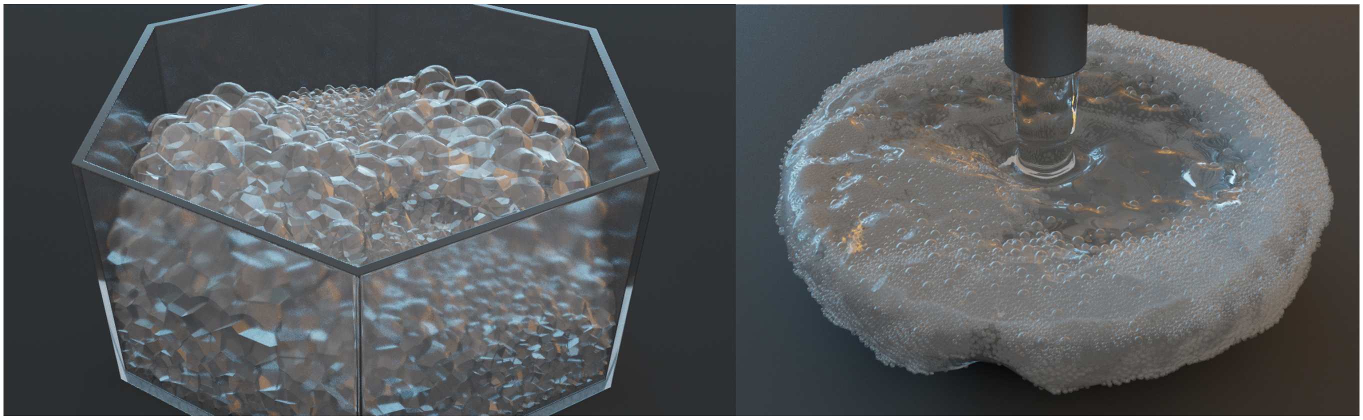

Fig. 1. Our new hybrid Lagrangian/Eulerian solver is capable of simulating mesoscale bubbles as a continuous medium (left), while also managing two-way coupling between disparate materials seamlessly (right). In contrast to traditional MPM, our material points are represented as volume-constrained discretized power diagrams, offering an accurate geometric depiction of mesoscale foam bubbles.

We present a novel hybrid Lagrangian/Eulerian method for simulating inelastic flows that generates high-quality particle distributions with adaptive volumes. At its core, our approach integrates an updated Lagrangian time discretization of continuum mechanics with the Power Particle-In-Cell geometric representation of deformable materials. As a result, we obtain material points described by optimized density kernels that precisely track the varying particle volumes both spatially and temporally. For efficient CFL-rate simulations, we also propose an implicit time integration for our system using a non-linear Gauss-Seidel solver inspired by X-PBD, viewing Eulerian nodal velocities as primal variables. We demonstrate the versatility of our method with simulations of mesoscale bubbles, sands, liquid, and foams.

CCS Concepts: • **Computing methodologies** → **Physical simulation**.

Additional Key Words and Phrases: Material point method, hybrid solvers, Particle-In-Cell methods, optimal transport.

ACM Reference Format:
Ziyin Qu, Minchen Li, Yin Yang, Chenfanfu Jiang, and Fernando de Goes. 2023. Power Plastics: A Hybrid Lagrangian/Eulerian Solver for Mesoscale Inelastic Flows . *ACM Trans. Graph.* 42, 6, Article 242 (December 2023), 11 pages. https://doi.org/10.1145/3618344

1 INTRODUCTION

Hybrid Lagrangian/Eulerian methods have been investigated extensively in computer graphics for simulating a diverse range of physical phenomena. In a typical hybrid solver, Lagrangian particles track the material motion, while forces are computed using a background Eulerian grid. A notable example of such numerical schemes is the Material Point Method (MPM) [Sulsky et al. 1995], which has demonstrated remarkable success in simulating granular and inelastic materials [Jiang et al. 2016]. While MPM works well for capturing the material's macroscale behavior approximated as a continuum, it becomes less applicable when a large number of coupled and deforming objects, like bubble foams, need to be taken care of, as existing approaches struggle to represent the intrinsic geometry of individual material elements.

In this paper, we present a new hybrid Lagrangian/Eulerian solver for the simulation of inelastic flows featuring mesoscale bubbles (Figure 1), i.e., structures that have already reached the Plateau's equilibrium, where their shapes can differ in size over time, but

Authors' addresses: Ziyin Qu, ziyinq@seas.upenn.edu, University of Pennsylvania, USA and University of California Los Angeles, USA; Minchen Li, minchen@math.ucla.edu, University of California Los Angeles, USA and Carnegie Mellon University, USA; Yin Yang, yin.yang@utah.edu, University of Utah, USA; Chenfanfu Jiang, cffjiang@math.ucla.edu, University of California Los Angeles, USA; Fernando de Goes, fernando@pixar.com, Pixar Animation Studios, USA.

0730-0301/2023/12-ART242 $15.00
https://doi.org/10.1145/3618344

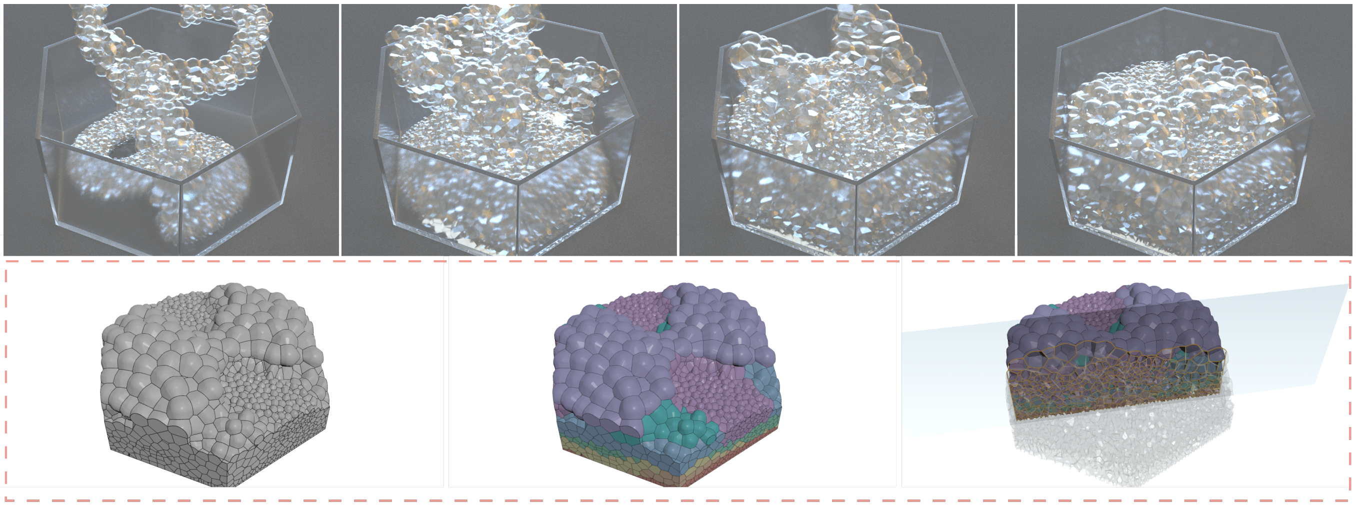

Fig. 2. We simulate dropping SIGGRAPH letter-shaped bubbles into a container, with each letter composed of bubbles of varied sizes and each bubble represented by a single particle. This experiment showcases the ability of our scheme to simulate the viscoplastic dynamics of bubbles, while maintaining the correct volume for each bubble. We further illustrate the geometric fidelity of our method by visualizing the underlying geometry of the bubbles, assigning each letter a unique color. The cross-section of this configuration is depicted in the bottom-right image.

their dynamics can still be modeled as a continuum [Stewart and Hilgenfeldt 2023]. Our approach builds upon the Power Particle-In-Cell Method [Qu et al. 2022], which computes particle-grid transfer weights through volume-constrained density kernels. We employ these transfer weights to enhance the MLS-MPM formulation [Hu et al. 2018] with basis functions that conform to particle volumes, thus removing the particle-per-cell limit common in prior MPM solvers. Additionally, we exploit the updated Lagrangian discretization provided by MLS-MPM in order to adapt the particle volumes and their associated transfer weights to the material deformation. To speedup performance, we also propose an implicit time-integrator using a non-linear Gauss-Seidel solver inspired by X-PBD [Macklin et al. 2016]. As a result, we obtain a MPM extension of the Power Particle-In-Cell framework well-suited for simulating mesoscale inelastic flows, hence the name *Power Plastics*.

In summary, the technical contributions of our work are:

- A hybrid Lagrangian/Eulerian solver capable of capturing the geometry of both macroscale and mesoscale materials.
- An extension of the Power Particle-In-Cell Method [Qu et al. 2022] that computes transfer weights adapted to time-evolving particle volumes estimated based on an updated Lagrangian discretization of inelastic deformations.
- An extension of MLS-MPM [Hu et al. 2018] that incorporates power weights, thus generating high-quality particle distributions free of any particle-per-cell restrictions.
- An implicit solver similar to X-PBD for faster time integration of inelastic flows within a MPM simulation.

2 RELATED WORK

Before presenting our contributions, we review prior work that are closely related to our approach, focusing our discussion on methods for foam simulation, hybrid solvers, and time integration.

2.1 Foam, bubble, and thin film simulation

Simulating foam, bubbles, and thin films has posed significant challenges to physics-based solvers due to their intricate geometric and dynamic properties. In graphics, various methods have been devised to simulate bubble foams contingent upon two material factors: scale and wetness. Particle-based approaches are commonly employed for small bubbles emulating the interplay between foam and fluid through buoyancy, drag, and cohesion [Bender et al. 2018; Busaryev et al. 2012; Hong et al. 2008; Wretborn et al. 2022]. Dense foams characterized by microscopic bubbles have been simulated as a non-Newtonian continuous material [Ram et al. 2015; Yue et al. 2015]. Large bubbles akin to soap films require explicit modelling of topological changes to aptly capture the progression of bubble merging and splitting [Da et al. 2015; Deng et al. 2022; Ishida et al. 2020, 2017; Zhu et al. 2014]. Our work targets instead the simulation of mesoscale bubbles, which describe volume-varying small structures that have attained a state of Plateau's equilibrium but can still be modeled as a continuum [Stewart and Hilgenfeldt 2023].

2.2 Hybrid Lagrangian/Eulerian methods

Hybrid solvers have found extensive application in graphics for the simulation of various phenomena. For instance, the Fluid Implicit Particle (FLIP) method [Brackbill and Ruppel 1986; Zhu and Bridson 2005] was introduced for the simulation of incompressible fluids with low dissipation. The Material Point Method (MPM) [Sulsky et al. 1995], on the other hand, has been successfully applied to the simulation of inelastic materials, including snow [Stomakhin et al. 2013], sand [Klár et al. 2016], and foam [Ram et al. 2015; Yue et al. 2015], while also handling substantial topological changes [Wolper et al. 2020, 2019]. The work of Hu et al. [2018] offered a reinterpretation of the MPM discretization through moving least squares shape functions, which served as a generalization of Affine

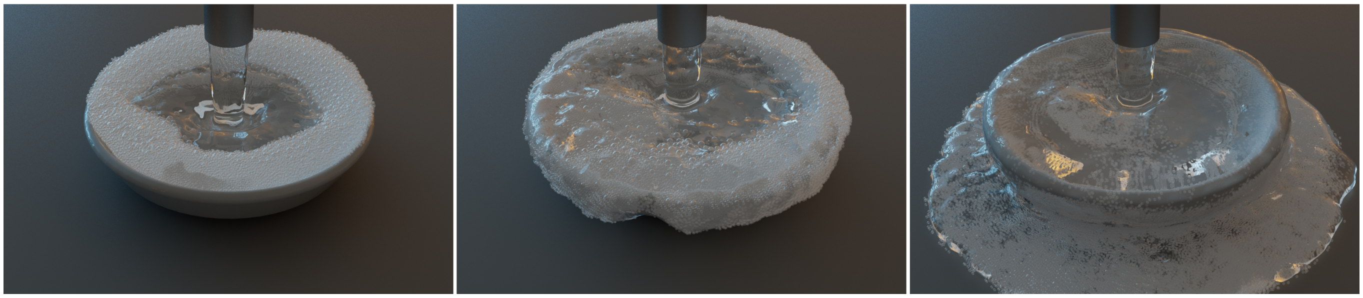

Fig. 3. We illustrate the two-way coupling of liquid and bubble foam by simulating fluid flushing bubble foams of varying sizes inside a bowl. Our method facilitates a natural interaction between the liquid and the bubbles, ensuring the bubbles float atop the liquid while maintaining their realistic geometry.

Particle-In-Cell [Jiang et al. 2015] and Polynomial Particle-In-Cell [Fu et al. 2017] methods. Despite their proficiency, MPM approaches tend to fall short for simulating mesoscale bubbles, where maintaining individual bubble geometry is crucial, due to particle clumping and volume drifting produced by existing solvers. Our approach draws inspiration from the Power Particle-In-Cell Method [Qu et al. 2022], which treats particles as a discretized power diagram, making it a suitable representation for the geometry of mesoscale bubbles. While Qu et al. [2022] focused on the simulation of incompressible fluids that keeps particle volumes constant in time, we instead account for dynamic particle volumes by simulating bubble foams as a continuous inelastic material through a modified MLS-MPM solver.

2.3 Optimization-based time integration

To avoid CFL restrictions on time-step sizes and achieve efficient simulation of elastic and inelastic solids under large deformations, the variational form of implicit time integration is often applied. In the work of Gast et al. [2015], implicit time integration was recast as a minimization problem for the simulation of hyperelastic materials with Rayleigh damping and collision penalties. Li et al. [2019] and Wang et al. [2020] implemented this optimization using a quasi-Newton strategy combined with domain decomposition and hierarchical preconditioning, respectively. The work of Bouaziz et al. [2014] proposed instead Projective Dynamics, which reformulates the implicit time integration into a local-global alternating minimization. This approach was later applied to simulate hyperelastic materials [Liu et al. 2017] as well as frictional contacts [Lan et al. 2022; Ly et al. 2020]. Narain et al. [2016] and Overby et al. [2017] extended the Projective Dynamics framework to a more general form of ADMM, which was then combined with MPM to simulate elastoplastic and viscoelastic phenomena [Fang et al. 2019]. Alternatively, position-based dynamics (PBD) [Müller et al. 2007] formulates various phenomena as constraints and optimizes these constrained systems in a Gauss-Seidel fashion [Bender et al. 2015, 2014]. In our work, we approach implicit time integration following on X-PBD [Macklin et al. 2016], an extension of PBD that captures accurate constitutive behaviors without sacrificing performance.

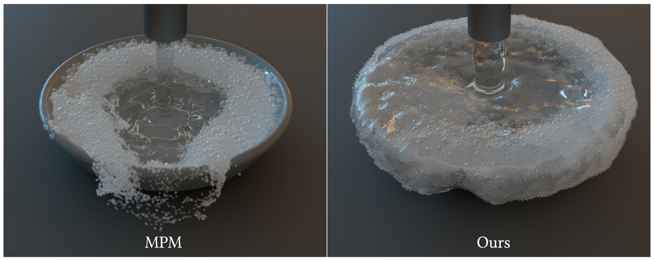

Fig. 4. We simulate flushing bubbles using MPM and our scheme. MPM is unable to capture the varying sizes of different foam bubbles, the insufficient PPC causes inaccurate physical behavior, where small foam bubbles leaking through and fluid shows obvious volume loss.

3 CONSTITUTIVE MODELS

In this section, we discuss our choice of constitutive model for mesoscale inelastic materials, and also recap other common constitutive models (e.g., sand and fluid) which will be used in our experiments. In the work of Busaryev et al. [2012], mesoscale bubbles were approximated using power diagrams with spring forces. While yielding results with both geometric and dynamic fidelity, this method was not feasible for modeling two-way coupling between foam bubbles and other materials. Moreover, this scheme relied heavily on the number of bubbles, making its scalability uncertain. Conversely, the work of Yue et al. [2015] represented dense foams as a non-Newtonian continuum conforming to the Herschel-Bulkley viscoplastic model. This formulation provided straightforward control over the foam dynamics, allowing it to exhibit either solid-like or fluid-like characteristics. The use of a solver based on continuum mechanics also made the integration of different materials seamless, while also facilitating two-way coupling. However, this approach can only approximate the macroscopic behavior of dense foams, failing to capture small geometric structures. In our work, we propose to simulate mesoscale bubbles by reconciling the geometric representation of bubbles via power diagrams proposed by Busaryev et al. [2012] with the dynamics of a dense foam through a continuous inelastic material similar to Yue et al. [2015].

Deformation Gradient: Given a deformation map $\mathbf{x} = \phi(\mathbf{X})$ which maps the continuum body from the reference configuration $\mathbf{X}$ to the deformed state $\mathbf{x}$, the deformation gradient is defined as $\mathbf{F}=\partial\phi/\partial\mathbf{X}$. As shown by Simo and Hughes [2006], the deformation gradient can be decomposed into $\mathbf{F} = \mathbf{F}_E\mathbf{F}_P$, where $\mathbf{F}_E$ is the part that generates elastic response versus $\mathbf{F}_P$ that indicates the plastic deformation.

Additionally, we use $\mathbf{F}_E = \boldsymbol{U}\Sigma\boldsymbol{V}^T$ to denote the singular value decomposition of the elastic part of the deformation gradient.

Compressible Fluids: We adopt the constitutive model for fluids proposed by Xie et al. [2023], which uses the energy density function

$$\psi(\mathbf{F}) = \frac{\lambda}{2}(J-1)^2, \tag{1}$$

where $J = \det(\mathbf{F})$ is the volume change induced by the deformation, and λ is the Lamé parameter.

Elasticity: For the elastic models of sand and foam, we adhere to the methodology of Klár et al. [2016] and employ the St. Venant-Kirchhoff (StVK) constitutive model in order to suppress unwanted volume fluctuations. The strain energy density function can then be expressed as

$$\psi(\mathbf{F}_E) = \mu\,\mathrm{tr}((\ln\Sigma)^2) + \frac{\lambda}{2}(\mathrm{tr}(\ln\Sigma))^2, \tag{2}$$

where μ and λ are Lamé parameters.

Plasticity: To quantify plasticity, one need to project the elastic stress onto the yield surface such that the strain satisfies the desired plastic flow. We first define the Hencky strain as $\boldsymbol{\epsilon} = \frac{1}{2}\ln(\mathbf{F}_E\mathbf{F}_E^T)$. Following Li et al. [2022b], we make use of the Hencky strain to write the von Mises yield condition as:

$$y(\boldsymbol{\tau}) = \|\hat{\boldsymbol{\tau}}\|_F - \sqrt{\frac{2}{3}}\sigma_Y \leq 0, \tag{3}$$

where σ_Y is the prescribed yield stress constant and $\hat{\boldsymbol{\tau}} = \boldsymbol{\tau} - \frac{1}{3}\,\mathrm{tr}(\boldsymbol{\tau})\mathbf{I}$ is the deviatoric part of Kirchoff stress $\boldsymbol{\tau} = 2\mu\boldsymbol{\epsilon} + \lambda\,\mathrm{tr}(\boldsymbol{\epsilon})\mathbf{I}$.

Herschel-Bulkley model: As described by Yue et al. [2015], the Herschel-Bulkley model quantifies the yield excess as

$$\gamma(\boldsymbol{\tau}) = \max\left(0, y(\boldsymbol{\tau})/\eta\right)^{1/h}, \tag{4}$$

where η denotes the viscosity coefficient. Here, the cases of $h < 1$ and $h > 1$ characterize shear-thinning and shear-thickening materials, respectively, while $h = 1$ recovers viscoplastic models, and $\eta = 0$ recovers perfect plasticity. We refer to Klár et al. [2016]; Yue et al. [2015] for the details on how to implement the plasticity projection.

4 DISCRETIZATION

We now describe the discretization of our system so that volume-varying materials like mesoscale bubbles can be resolved both geometrically and dynamically within a hybrid Lagrangian/Eulerian framework. We start by presenting our geometric representation of Lagrangian particles based on the Power Particle-In-Cell Method [Qu et al. 2022]. We then show how to incorporate the Power Particle representation into the MLS-MPM formulation [Hu et al. 2018], which offers deformation gradient estimates per particle. At last, we make use of these deformation gradients to update particle volumes and compute forces using any choice of constitutive model.

4.1 Volume-aware geometric representation

For modeling mesoscale bubbles, we need a geometric representation capable of capturing the volume characteristics of bubble foams. In the work of Busaryev et al. [2012], small bubbles were discretized using power diagrams based on the observation that they are much less deformable than larger bubbles due to the predominance of surface tension, but they can still vary in size. While power diagrams provide the adaptive volumes we seek for, the work of Busaryev et al. [2012] limited the interaction between bubbles to spring forces decoupled from any other material. Power diagrams were later leveraged by the Power Particles technique [de Goes et al. 2015] in order to define a Lagrangian discretization of incompressible fluids with precise volume control. However, its repetitive optimization of power diagrams to enforce volumes becomes costly as the particle count escalates. Additionally, dealing with free surface necessitates either air particle sampling [de Goes et al. 2015] or polytope clipping [Lévy 2021], which adds further complexity. More recently, the Power Particle-In-Cell Method [Qu et al. 2022] reframed the construction of Power Particles as a regularized optimal transport problem, which enables faster computations free of any explicit cell clipping. Within this framework, Power Particles can be regarded as volume-constrained density kernels that resemble the formation of power diagrams rasterized over a background grid. Importantly, these density kernels can be retrofitted into volume-preserving transfer weights, thus integrating the precise volume control from Power Particles into hybrid Lagrangian/Eulerian solvers. We thus adopt the Power Particle-In-Cell Method as our geometric representation of mesoscale bubbles.

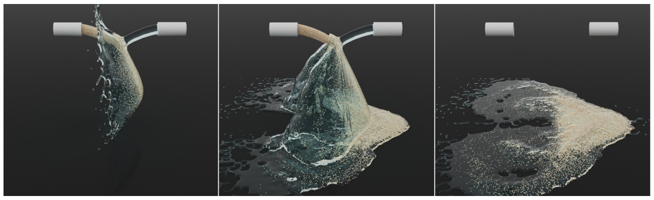

Fig. 5. Our method naturally handles the strong two-way coupling of different materials, such as simulating water and sand jets in this example. Moreover, our scheme generates smooth fluid free surface due to evenly spaced particle distribution.

Volume-constrained power diagram: Following Qu et al. [2022], we review next the Power Particle-In-Cell Method, which approximates the construction of power diagrams with prescribed particle volumes cast as an optimal transport problem. Suppose we have n_p particles with position $\mathbf{x}_p$ and volume V_p, and an auxiliary transportation grid (t-grid) of size n_j with cell position $\mathbf{x}_j$ and capacity V_j. For conciseness, we assume a closed container with no free surface, and point the reader to the work of Qu et al. [2022, §6] for details on how to account for free surface. Our goal is to find the transportation plan that minimizes the cost of moving volume between particles and t-grid cells, while enforcing volume constraints:

$$\min_{\{T_{pj}\}} \textstyle\sum_{p,j} T_{pj}\|\mathbf{x}_p - \mathbf{x}_j\|^2 + \varepsilon \sum_{p,j} \mathcal{H}(T_{pj}) \tag{5a}$$

$$\text{s.t.} \begin{cases} \sum_p T_{pj} = V_j & \forall j, \\ \sum_j T_{pj} = V_p & \forall p, \end{cases} \tag{5b}$$

where T_{pj} indicates the amount of volume transported from particle p to t-grid cell j, ε is the regularization amount controlling the impact range of each particle, and $\mathcal{H}(x) = x\log(x) - x$ is the entropy function. This optimization has a unique solution of the form

$$T_{pj} = \exp(r_j/\varepsilon)\exp(r_p/\varepsilon)\exp(-\|\mathbf{x}_p - \mathbf{x}_j\|^2/\varepsilon) = s_j s_p K_{pj}. \tag{6}$$

Here, r_p and r_j are the Lagrange multipliers for the particle and grid volume constraints, s_p and s_j are the corresponding scaling amount, and K_{pj} is the Gaussian kernel between particle p and t-grid cell j. Therefore, minimizing Eq. (5) amounts to finding a scaling per particle s_p and another scaling s_j per t-grid cell, which can be efficiently solved via the Sinkhorn's algorithm [Cuturi 2013].

Power weights: Once we obtained the transportation plan $\{T_{pj}\}$, we can construct a density kernel $\chi_p^\varepsilon : \Omega \to [0, 1]$ indicating the spatial occupancy of each particle via

$$\chi_p^\varepsilon(\mathbf{x}) = \frac{\exp((r_p - \|\mathbf{x}_p - \mathbf{x}\|^2)/\varepsilon)}{\sum_q \exp((r_q - \|\mathbf{x}_q - \mathbf{x}\|^2)/\varepsilon)}. \tag{7}$$

Note that $\sum_p \chi_p^\varepsilon(\mathbf{x}) = 1$ and, when evaluating the density kernel at t-grid cell j, we have $\chi_p^\varepsilon(\mathbf{x}_j) = T_{pj}/\sum_q T_{qj} = T_{pj}/V_j$. Moreover, using t-grid cells as quadrature points, the integration of the density kernel recovers the volume of its associated particle:

$$\int_\Omega \chi_p^\varepsilon(\mathbf{x})d\mathbf{x} \approx \sum\nolimits_j \chi_p^\varepsilon(\mathbf{x}_j)V_j = \sum\nolimits_j T_{pj} = V_p. \tag{8}$$

We can further define the centroid $\mathbf{c}_p$ for each particle p as

$$\mathbf{c}_p = \frac{1}{V_p}\int_\Omega \chi_p^\varepsilon(\mathbf{x})\mathbf{x}d\mathbf{x} \approx \frac{1}{V_p}\sum\nolimits_j T_{pj}\mathbf{x}_j. \tag{9}$$

Up to now, we only constructed the transportation between particles and t-grid, which is an auxiliary grid separated from the actual simulation grid (s-grid). We define the power weights as the transfer between particles and s-grid cells computed via

$$w_{pi} = \frac{1}{V_p}\int_\Omega \chi_p^\varepsilon(\mathbf{x})N_i(\mathbf{x})\ d\mathbf{x} \approx \frac{1}{V_p}\sum\nolimits_j T_{pj}N_i(\mathbf{x}_j), \tag{10}$$

where $N_i(\mathbf{x})$ is an interpolation function centered at each s-grid cell i with properties $N_i(\mathbf{x}) \geq 0$, $\sum_i N_i(\mathbf{x}) = 1$, and $\sum_i N_i(\mathbf{x})\mathbf{x}_i = \mathbf{x}$. For efficiency and simplicity, we set $N_i(\mathbf{x})$ to a linear kernel. Finally, one can verify that the power weights satisfy the following properties:

$$\begin{cases} w_{pi} \geq 0, \\ \sum_i w_{pi} = 1, \\ \sum_i w_{pi}\mathbf{x}_i = \mathbf{c}_p. \end{cases} \tag{11}$$

It is worth noticing that $\sum_i w_{pi}\mathbf{x}_i$ returns the centroid $\mathbf{c}_p$, in sharp contrast to typical nodal interpolants which return $\mathbf{x}_p$ instead.

4.2 Dynamic volume updates

To adjust the size of the mesoscale bubbles over time, we need to estimate the volume change induced by the material deformation. Similar to prior MPM solvers, we assign each particle p to a deformation gradient matrix $\mathbf{F}_p$ and compute its time update via:

$$\frac{D\mathbf{F}}{Dt} = (\nabla\mathbf{v})\mathbf{F}. \tag{12}$$

Using an updated Lagrangian discretization, we can approximate the evolution of deformation gradient within a time-step Δt from time n to $n + 1$ via:

$$\mathbf{F}_p^{n+1} = \left(\mathbf{I} + \Delta t\frac{\partial\mathbf{v}^{n+1}}{\partial\mathbf{x}}(\mathbf{x}_p^n)\right)\mathbf{F}_p^n, \tag{13}$$

which requires the evaluation of velocity gradient $(\partial\mathbf{v}^{n+1}/\partial\mathbf{x})(\mathbf{x}_p^n)$.

Typically, hybrid Lagrangian/Eulerian approaches like MPM evaluate the velocity gradient using the derivatives of the nodal shape functions over the simulation grid. We instead avoid nodal derivatives by constructing moving least squares (MLS) shape functions based on our volume-aware power weights. Our derivation follows a schematic similar to Hu et al. [2018], computing the MLS shape function for the s-grid node i nearby a particle p at time n via:

$$\Phi_i^n(\mathbf{x}) = w_{pi}^n\,\mathbf{P}^T(\mathbf{x} - \mathbf{c}_p^n)\,\mathbf{M}^{-1}\,\mathbf{P}(\mathbf{x}_i - \mathbf{c}_p^n), \tag{14}$$

where $\mathbf{M} = \sum_i w_{pi}^n\mathbf{P}(\mathbf{x}_i - \mathbf{c}_p^n)\mathbf{P}^T(\mathbf{x}_i - \mathbf{c}_p^n)$ is the momentum matrix for particle p, and $\mathbf{P}(\mathbf{x})$ is a subspace of polynomial bases. Note that, in comparison to Hu et al. [2018], our MLS shape functions include power weights instead of B-Splines and are centered at the particle centroid $\mathbf{c}_p$ instead of $\mathbf{x}_p$.

We can now approximate the material velocity $\mathbf{v}^{n+1}(\mathbf{x})$ at time $n + 1$ evaluated at any point $\mathbf{x}$ located nearby the particle p as:

$$\mathbf{v}^{n+1}(\mathbf{x}) = \sum\nolimits_i \Phi_i^n(\mathbf{x})\,\mathbf{v}_i^{n+1}, \tag{15}$$

where $\mathbf{v}_i^{n+1}$ is the discretized velocity at a s-grid cell i. In our implementation, we restrict $\mathbf{P}(\mathbf{x})$ to linear basis for simplicity, which combined with Eq. (11) reduces the velocity derivative to

$$\frac{\partial\mathbf{v}^{n+1}}{\partial\mathbf{x}}(\mathbf{x}_p^n) = \mathbf{A}_p^{n+1} = \mathbf{B}_p^{n+1}(\mathbf{D}_p^n)^{-1}, \tag{16}$$

formed by the matrices

$$\begin{cases} \mathbf{D}_p^n = \sum_i w_{pi}^n(\mathbf{x}_i - \mathbf{c}_p^n)(\mathbf{x}_i - \mathbf{c}_p^n)^T, \\ \mathbf{B}_p^{n+1} = \sum_i w_{pi}^n\mathbf{v}_i^{n+1}(\mathbf{x}_i - \mathbf{c}_p^n)^t. \end{cases} \tag{17}$$

Note that our linear version of the MLS shape functions led to the same matrices used by the Power APIC transfer [Qu et al. 2022, §5].

Using linear MLS shape functions, the update of the deformation gradient in Eq. (18) boils down to

$$\mathbf{F}_p^{n+1} = \left(\mathbf{I} + \Delta t\mathbf{A}_p^{n+1}\right)\mathbf{F}_p^n. \tag{18}$$

Moreover, denoting the initial particle volume as V_p^0, we can finally evolve the particle volume in time via

$$V_p^{n+1} = \det(\mathbf{F}_p^{n+1})V_p^0, \tag{19}$$

which is then used as particle constraints in Eq. (5b) to adapt the power weights in the next simulation time-step.

4.3 Force computation

To simulate a continuous material, one need to update the velocity field $\mathbf{v}$ in time through the momentum equation:

$$\rho\frac{D\mathbf{v}}{Dt} = \nabla\cdot\boldsymbol{\sigma} + \rho\mathbf{g}, \tag{20}$$

where ρ is the density, $\mathbf{g}$ is gravity, $\boldsymbol{\sigma}$ is the Cauchy stress induced by the constitutive model, and $D/Dt = \partial/\partial t + \mathbf{v}\cdot\nabla$ is material derivative.

Equipped with MLS shape functions from Eq. (14), we can construct test functions $\mathbf{q}(\mathbf{x}) = \sum_i \Phi_i(\mathbf{x})\mathbf{q}_i$, and then derive an updated Lagrangian weak form discretization of Eq. (20). After applying

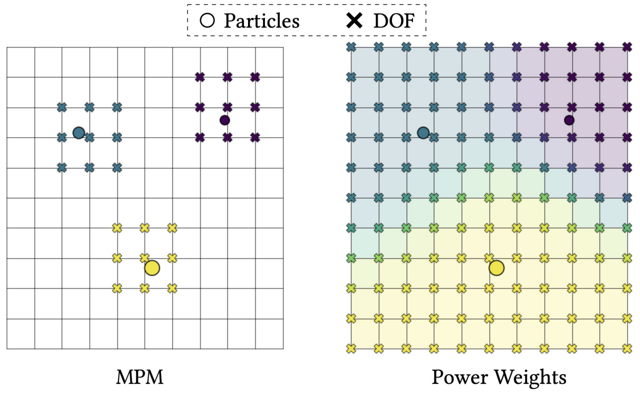

Fig. 6. In this example, we consider particles sampled inside a closed container with their combined volume matching the container's volume. In MPM with quadratic B-spline (left), the grid DOF is only defined locally near particles with a fixed window, despite the fact that each particle represents a different volume. In contrast, our power weights determine the grid DOF according to particles actual sizes, as indicated by the pseudo-colored particle influences (right). As a result, our scheme allows a single particle to span multiple cells, removing any particle-per-cell restriction.

Dirichlet boundary conditions and omitting the gravity term for conciseness, we have:

$$\frac{1}{\Delta t}\int_{\Omega}\rho(\mathbf{x})\,(\hat{\mathbf{v}}_\alpha(\mathbf{x})-\mathbf{v}_\alpha(\mathbf{x}))\,\mathbf{q}_\alpha(\mathbf{x})d\mathbf{x}=-\int_{\Omega}\mathbf{q}_{\alpha,\beta}(\mathbf{x})\sigma_{\alpha\beta}(\mathbf{x})d\mathbf{x},\quad(21)$$

where Ω is the simulation domain, Δt is the time-step size, $\mathbf{v}$ is the previous velocity, and $\hat{\mathbf{v}}$ is the force updated velocity. Here, we use tensor notation to indicate a vector coordinate $\mathbf{v}_\alpha$, a matrix entry $\sigma_{\alpha\beta}$, and the partial derivative $\mathbf{q}_{\alpha,\beta}=\partial\mathbf{q}_\alpha/\partial\mathbf{x}_\beta$.

Following the same derivation steps of Hu et al. [2018, §3.3.5], we can expand the right-hand side of Eq. (21) and compute the internal force $\mathbf{f}_i$ discretized at a s-grid cell i as

$$\mathbf{f}_i=-\sum\nolimits_p w_{pi}V_p^0\mathbf{D}_p^{-1}\frac{\partial\Psi}{\partial\mathbf{F}}(\mathbf{F}_p^n)\mathbf{F}_p^{nT}(\mathbf{x}_i^n-\mathbf{c}_p^n).\quad(22)$$

Observe, once again, that our formulation differs from Hu et al. [2018] due to the use of the centroid $\mathbf{c}_p$ as quadrature points and power weights w_{pi} as weighting function. Moreover, notice that Eq. (22) includes the contribution of the desired constitutive model Ψ.

5 IMPLICIT TIME INTEGRATOR

Explicit time integration can be prohibitive for simulating stiff systems due to CFL time-step restrictions. Similar to Gast et al. [2015], we formulate the implicit backward Euler system as an unconstrained minimization with the velocity as the primal variable:

$$\min_{\mathbf{v}}\frac{1}{2}\sum\nolimits_i m_i\|\mathbf{v}_i-\hat{\mathbf{v}}_i\|^2+\sum\nolimits_p V_p^0\psi(\mathbf{F}_p^{n+1}),\quad(23)$$

where $\hat{\mathbf{v}}_i$ is the velocity updated by external forces at the s-grid cell i with lumped mass m_i. This optimization can be robustly solved using Newton's method with line search [Gast et al. 2015], or ADMM [Narain et al. 2016]. However, frequently rebuilding the global systems due to topology changes can be computationally expensive.

Instead, we draw inspiration from Macklin et al. [2016] by treating the non-linear elastic energy potential as constraint per particle:

$$V_p^0\psi(\mathbf{F}_p^{n+1})=\frac{1}{2\alpha}\Big[C_p(\mathbf{F}_p^{n+1})\Big]^2,\quad(24)$$

where $\alpha=1/(2V_p^0)$ is a shorthand for the compliance coefficient and the constraint functions are $C_p(\mathbf{F}_p^{n+1})=\sqrt{\psi_p(\mathbf{F}_p^{n+1})}$. Following the same linearization as in Macklin et al. [2016], we obtain similar update rules for the Lagrange multiplier λ_p and s-grid velocity $\mathbf{v}_i$:

$$\Delta\lambda_p^k=\frac{-C_p(\mathbf{F}_p^k)-(\alpha/\Delta t^2)\lambda_p^k}{\nabla C_p\mathbf{M}^{-1}\nabla C_p^T+(\alpha/\Delta t^2)},\quad(25a)$$

$$\Delta\mathbf{v}_i^k=\mathbf{M}^{-1}\nabla C_p(\mathbf{F}_p^n)^T\Delta\boldsymbol{\lambda}/\Delta t,\quad(25b)$$

where computing ∇C_p involves a particle-grid transfer and $\mathbf{M}$ is the particle momentum matrix. Our implicit optimization thus emulates a Gauss-Seidel style solver, alleviating the need for constructing a global matrix. It also allows a parallel implementation by utilizing graph-coloring algorithms, as suggested by Hu et al. [2018] and Bender et al. [2017]. Boundary conditions are enforced at the end of each solver iteration. We further point out that our solver can be directly employed in other hybrid methods too. All the simulations in our paper are performed using our implicit solver, with approximately 10 iterations per time-step.

6 IMPLEMENTATION DETAILS

In this section, we outline the main steps of our hybrid solver, discuss some practical considerations that accelerate the computations of our volume-adaptive extension of the Power Particle-In-Cell Method, and finally detail how to render bubbles as power diagrams.

6.1 Algorithm

The overall structure of our algorithm resembles a typical MPM solver, which can be summarized as follows:

(1) **Compute transportation plan:** Perform Sinkhorn's iterations to solve Eq. (5).
(2) **Compute power weights:** Aggregate transportation plan into power weights using Eq. (10).
(3) **Particle to grid:** The mass and velocity are transferred using Power APIC [Qu et al. 2022], which sets $m_i=\sum_p w_{pi}m_p$ and $m_i\mathbf{v}_i=\sum_p w_{pi}m_p\left(\mathbf{v}_p+\mathbf{A}_p(\mathbf{x}_i-\mathbf{c}_p)\right)$.
(4) **Update grid momentum:** Update the s-grid velocity $\hat{\mathbf{v}}_i$ with external forces and running our X-PBD implicit time integrator (§5) with internal forces computed via Eq. (22).
(5) **Grid to particle:** The particle velocity is transferred from the s-grid via $\mathbf{v}_p=\sum_i w_{pi}\hat{\mathbf{v}}_i$, and the matrices $\mathbf{B}_p$ and $\mathbf{D}_p$ are updated through Eq. (17).
(6) **Update particle deformation gradient:** Deformation gradient is updated per particle using Eq. (18).
(7) **Update particle plasticity:** The plasticity is projected according to the material's constitutive model.
(8) **Update particle volume:** The volume of each particle is resized using Eq. (19).
(9) **Particles advection:** Following Qu et al. [2022], particle positions are updated using $\mathbf{x}_p=\mathbf{c}_p+\Delta t\,\mathbf{v}_p$.

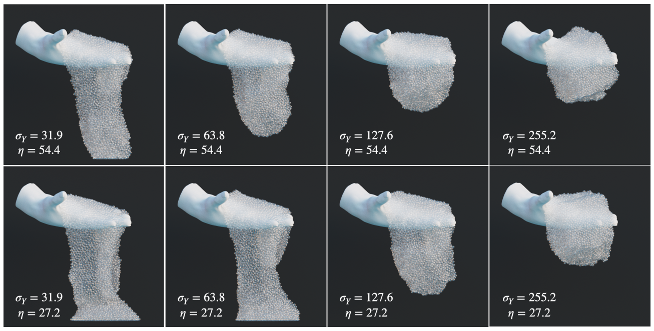

Fig. 7. We conducted simulations of bubbles falling onto a hand to examine the impact of various parameters. Utilizing the Herschel-Bulkley model and setting $h = 1.0$ (representing viscoplastic behavior), we progressively increased the yield stress and viscosity coefficient (σ_Y and η) from left to right and bottom to top respectively. These results display the variance in the bubbles' dynamic behavior as a function of these material parameters.

Comparing our framework with that of Qu et al. [2022], which is confined to incompressible fluid simulation, our approach introduces the robust coupling and topological changes of multiple materials through steps 4, 6, and 7. In contrast to prior MPM solvers, our implementation incorporates the geometric representation provided by Power Particles through steps 1, 2, 8, and 9. Importantly, our formulation enables a single large power parcel to span multiple s-grids cells, as illustrated in Figure 6, whereas in B-spline MPM each particle only embodies a fixed window of cells. This implies that, unlike MPM, our method circumvents the restriction of minimum particle-per-cell (PPC), improving its versatility and robustness. Finally, we point out that boundary conditions can be imposed exactly the same as in a typical MPM [Jiang et al. 2016], with wall collisions and external forces applied onto grids and particles.

6.2 Adaptive regularization

In the Sinkhorn's iterations that compute the transportation plan, the work of Qu et al. [2022] suggested a cutoff threshold based on the regularization coefficient ε in order to sparsify the particle-grid kernel by clamping K_{pj} to zero when $\|\mathbf{x}_p - \mathbf{x}_j\| > 3\sqrt{\varepsilon}$. However, we noticed in practice that using the same cutoff ε for particles of different volumes might cause convergence issues in the Sinkhorn's algorithm, especially on particles with large volumes because they require more iterations to expand their coverage.

In this work, we propose instead a base regularization coefficient $\varepsilon_{\min} = 2(\Delta x)^2$, where Δx is the t-grid cell size, and then adjust the regularization amount for each particle based on their radius R_p (computed from particle volume V_p by treating them as a sphere):

$$\varepsilon_p = \left(\frac{R_p}{R_{\min}}\right)^2 \varepsilon_{\min}, \tag{26}$$

where $R_{\min} = \min_p R_p$. Using this adaptive regularization, we assemble the sparse kernel matrix through:

$$K_{pj} = \begin{cases} \exp(-\|\mathbf{x}_p - \mathbf{x}_j\|^2/\varepsilon_p) & \text{if } \|\mathbf{x}_p - \mathbf{x}_j\| \le 3\sqrt{\varepsilon_p}, \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

This modification implies that particles with larger volumes end up more regularized and, consequently, with a kernel covering a broader region of influence over the t-grid. In our experiments, this strategy led to faster convergence of the Sinkhorn's solver.

6.3 Bubble rendering

To render bubbles with realism, one must represent the inner-surfaces between bubbles in addition to the material's free surface. However, the work of Qu et al. [2022] only discussed how to iso-contour the free surface from the Power Particles discretization. Initially, we considered extracting the inner-surfaces by iso-contouring an level-set from each particle's density kernel $\chi_p^\varepsilon(\mathbf{x})$ and computing their boolean against the free surface, but these meshing operations were too costly. Since the Lagrangian multipliers r_p generated by the Sinkhorn's algorithm are approximations of the weights used to form a power diagram [Qu et al. 2022, §4], we also tried to compute the inner-surfaces by constructing an unbounded power diagram using these approximated weights and then clipping the power cells against the free surface, but this was still computationally expensive. As a more practical alternative, our ultimate solution was to reuse the weight optimization proposed by de Goes et al. [2015] that constructs power diagrams by seeding the free surface with air particles. In our implementation, we perform this construction only once per frame as post-processing, and then render the facets shared by power cells as our inter-surfaces. All our images showing bubbles were rendered using this approach.

7 RESULTS AND DISCUSSION

We now present a series of experiments that showcase the quality and versatility of our method. The configurations and timing for all our examples are summarized in Table 1, clocked on a Intel Core

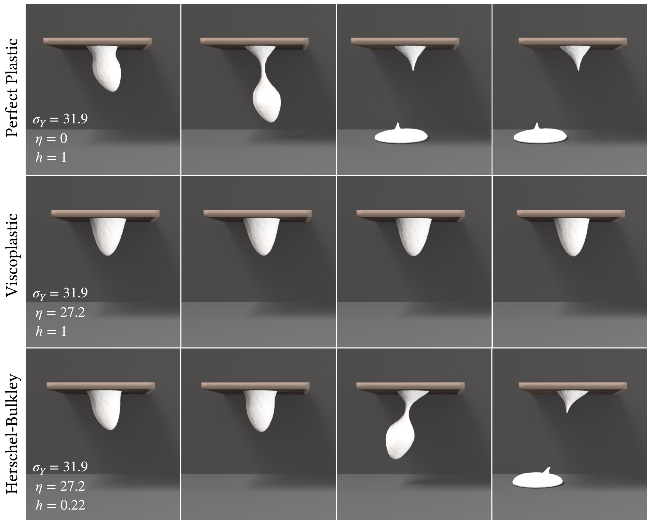

Fig. 8. To validate our continuum foam model, we reproduce the shaking foam example, showcasing perfect plastic, viscoplastic, and Herschel-Bulkley behaviors. Our results align with those by Yue et al. [2015], attesting to the correctness of our method.

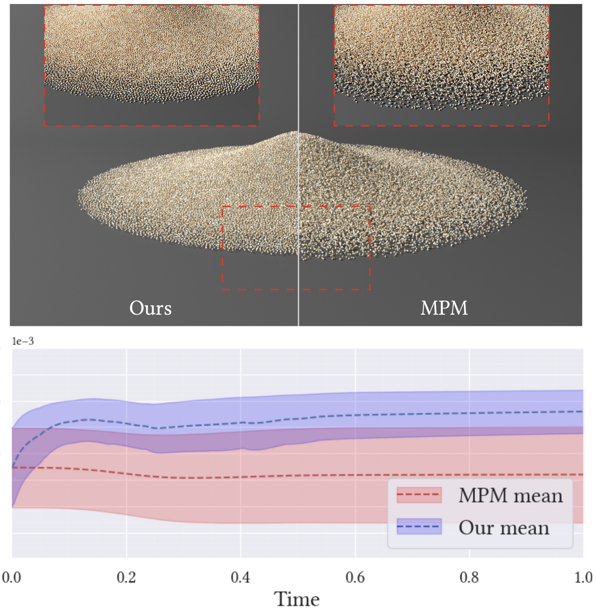

Fig. 9. In this example, we simulate a sand column collapse, comparing the results from our method (left) with MPM (right). Both approaches successfully produce the correct friction angle upon the sand settling. However, our method exhibits more uniform particle distribution, resolving voids and clumps typically caused by hybrid solvers. In the bottom image, we compare these results quantitatively by plotting the averaged distance between each particle and its nearest neighbor per simulation frame, with the shaded area indicating the respective standard deviations.

i7-9700 with 32GB memory. Our CPU-based implementation used OpenVDB [Museth et al. 2013] as our sparse grid data structure and Eigen [Guennebaud et al. 2010] as our linear algebra library.

Rayleigh-Taylor instability. In Figure 12, we test the Rayleigh-Taylor instability initialized with compressible fluids of different densities. The blue particles are set with a density of $\rho = 500\text{kg}/m^3$, whereas the red particles are initialized with a higher density of $\rho = 1000\text{kg}/m^3$. Although both MPM and our method can accurately capture the dynamic behavior, MPM exhibits significant volume loss as the simulation progresses. In sharp contrast, our method manages to maintain the fluid volume error within a configurable range, underscoring the improved volume conservation of our approach.

Volume-adaptive bubbles. In Figure 2, we show SIGGRAPH letter-shaped bubbles with varying sizes dropped into a container, each bubble represented by a single particle. Due to the incorporation of the power weights, our scheme delivers accurate results both dynamically and geometrically. In particular, our method is able to decouple the dependency between the particle count and the grid resolution, a typical requirement in MPM that forces at least 4 particles per cell (PPC) for 2D and 8 PPC for 3D. In our approach, a single particle can be sized several times larger or smaller than the grid spacing Δx, as displayed by Figure 6, thus offering a hybrid solver free of any the particle-per-cell restriction.

Two-way coupling. We also demonstrate the effective two-way coupling between different materials within our simulation framework, including water/bubbles in Figure 3 and water/sand in Figure 5. For the experiment in Figure 3, we initialized bubbles of various sizes within a bowl-shaped container and subsequently introduced a flow of fluid into it. The resulting bubbles successfully float atop the fluid, illustrating the physically plausible interaction between the two materials. Importantly, our method led to an improved representation of both the geometry and dynamics of foamy bubbles, as showcased by the comparison to a typical MPM solver in Figure 4. However, it is worth emphasizing that our method does not handle merging and splitting of bubbles, a feature we leave as future work.

Effect of parameters. To assess the accuracy of our method with inelastic flows, we replicate with our method the "shaking foam" example from Yue et al. [2015] in Figure 8. Adopting the exact material parameters delineated by Yue et al. [2015], we successfully emulate perfect plasticity, viscoplasticity, and shear-thinning effects, demonstrating the capability of our method to accurately reproduce the Herschel-Bulkley model. To further examine the influence of material parameters on bubble dynamics, specifically the yield stress σ_Y and the viscosity coefficient η, we also run the simulation of a hand holding bubbles in Figure 7. In these tests, we maintained $h = 1.0$ for the Herschel-Bulkley model. As expected, we observed that lower yield stress and viscosity values presented fluid characteristics, while larger values caused bubbles to behave more like solids.

Particle distribution. In Figure 9, we highlight the capability of our method to simulate sand, while concurrently maintaining an evenly-spaced particle distribution. In this test, we applied identical simulation parameters to both the MPM solver and our method, including time-step, grid resolution, material parameters, and the sand's friction angle. Interestingly, our particle distribution is less noisy compared to MPM, resembling a blue-noise pattern. This is expected since our advection step moves particles to their centroids. For a quantitative comparison, we also plotted the averaged distance between each particle and its nearest neighbor per frame, showing that our scheme has a 52% tighter variance range versus MPM, see Figure 9. It is also worth noting that our approach offers flexibility regarding particle distribution. If a non-uniform particle distribution is preferred, one can seamlessly adjust it by introducing a blending coefficient α during our advection computation. This coefficient blends the particle position $\mathbf{x}_p$ and the centroid position $\mathbf{c}_p$ so that $\mathbf{x}_p^{n+1} = \mathbf{c}_p^n + \alpha(\mathbf{x}_p^n - \mathbf{c}_p^n) + \mathbf{v}_p^n \Delta t$, effectively modulating the distribution to desired specifications.

Performance. To assess performance, we plot in Figure 11 the computation breakdown for an explicit versus an implicit implementation of our solver applied to the sand example from Figure 9. For the explicit case, our solver was constrained to a time-step of $\Delta t = 5.0 \times 10^{-4} s$ due to the CFL condition. Each time-step consumes 4.3 seconds, with the creation of the transportation plan and the assembly of power weights comprising 78% of the total time. In contrast, our X-PBD-based solver allowed a 10x increase in time-step, with each time-step taking 8.8 seconds and our implicit solver (with 10 iterations per time-step) emerging as the most costly step. We also include the timing comparisons against MPM in Table 1, our method is approximately 2.5x slower, and the extra overhead is mainly from the transportation plan computation.

Stability. To study the stability of our implicit solver, Figure 10 displays the simulation of a sand collapse computed by our implicit X-PBD solver with various iteration counts. Although our implicit solver generates qualitatively similar results under different iterations, the total iteration count might need to be increased as the resolution of the simulation grid gets finer or the material gets stiffer. Therefore, our implicit time integrator effectively alleviates the CFL time-step restriction compared to explicit schemes, but it is not unconditionally stable.

8 CONCLUSION

This paper presents *Power Plastics*, a novel hybrid Lagrangian/Eulerian solver for the simulation of mesoscale inelastic flows that extends the Power Particle-In-Cell Method [Qu et al. 2022] into a MLS-MPM framework [Hu et al. 2018]. The strength of our approach lies in its ability to effectively simulating volume-adapting discrete materials, including but not limited to foam bubbles, sand, and fluids. By doing so, our solver generates high-quality particle distributions with precise control of particle volumes. Moreover, our method avoids the need for traditional Newton-type solves by introducing an efficient implicit time integration based on X-PBD [Macklin et al. 2016].

For future work, there are several improvements and extensions to be investigated. Despite our solver's proficiency with volume-varying materials, computational load balancing can become a challenge if there is a substantial volume disparity among particles. Therefore, we are considering to exploit adaptive data structures, like octrees, to ameliorate this issue and enhance computational efficiency. We are also interested in physically based models for representing dynamic merging and splitting of bubbles within our framework. In addition to inelastic flows targeted by our current work, we would like to extend our solver to elastic materials, including hair, cloth, and flesh-like substances. Last but not least, our implementation computes the plasticity projection only once per time-step, but it would be meaningful to fully incorporate the plasticity projection during the time-stepping optimization, as previously done by Li et al. [2022a,c].

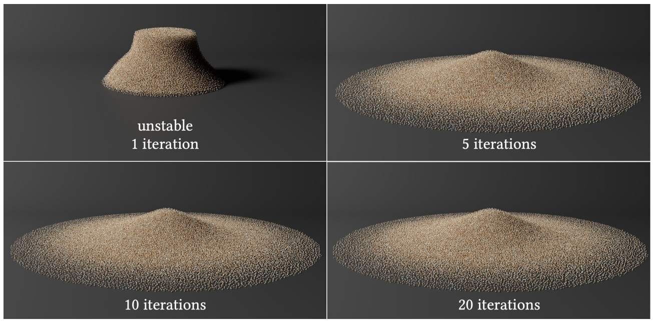

Fig. 10. We test the stability of our implicit time integrator using a sand collapse example. The simulation is unstable with a single iteration, but it stabilizes by adding more iterations, exhibiting qualitatively similar results.

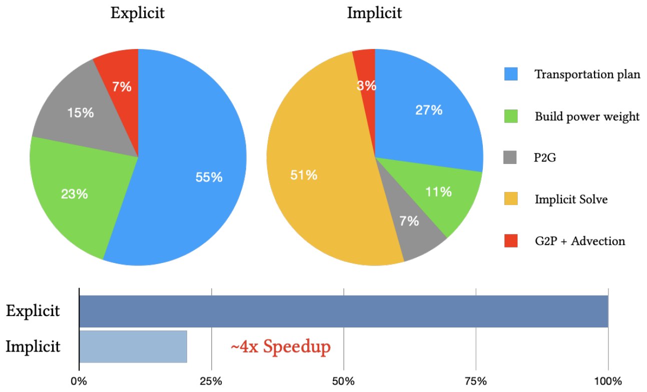

Fig. 11. We compare the performance of our explicit versus implicit time integration in the sand example from Figure 9. Although the implicit solve is a costly step, it enables a larger time-step allowing the implicit scheme to deliver approximately 4× speedup compared to the explicit approach.

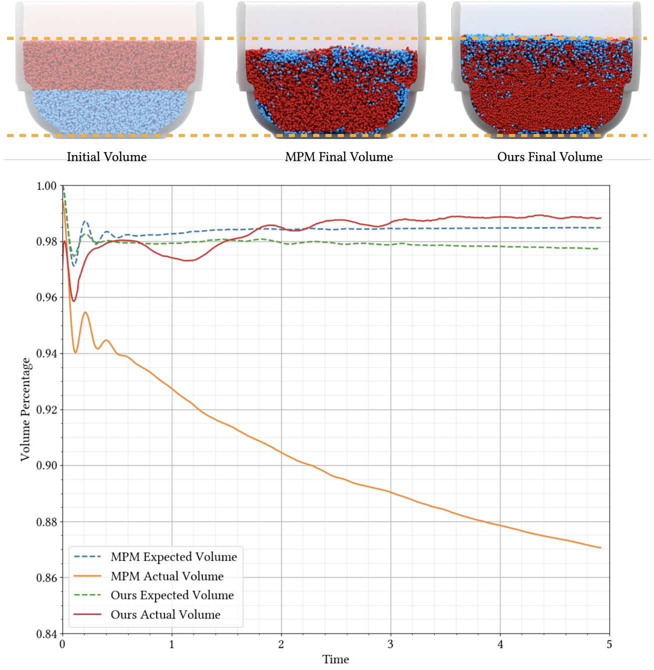

Fig. 12. To simulate Rayleigh-Taylor instability, we introduce fluids with two different densities inside a container; where the denser fluid is indicated in red and the lighter one in blue. The expected volume is calculated by summing up the volumes of individual particles, while the actual volume is determined by measuring the mesh constructed from these particles.

9 ACKNOWLEDGMENTS

We sincerely thank the anonymous reviewers for their valuebale feedback. This work is partially supported by NSF under the grant

Table 1. This table reports the parameters and timing for our experiments. The value Δt is the time-step size, Δx_s and Δx_t denote the grid width for the s-grid and t-grid, respectively. ρ is the density, $(E, \nu, \sigma_Y, \eta, h)$ are Young's modulus, Poisson's ratio, yield stress, viscosity coefficient, and Herschel-Bulkley shear characteristic parameter, respectively.

Example	N_p	Δt[s]	Δx_t[m]	Δx_s[m]	$\rho[kg/m^3]$	E[Pa]	ν	σ_Y[Pa]	η	h	sec/step(Ours)	sec/step(MPM)	performance	integration
Figure 2	9k	0.001	0.0025	0.005	77	10^5	0.3	63.8	54.0	1.0	3.8	-	-	implicit
Figure 3	300k	0.002	0.005	0.01	77/1000	10^5	0.4/0.3	63.8	27.0	1.0	21.5	7.8	0.36x	implicit
Figure 5	60k	0.001	0.002	0.004	1000/2000	$10^5/10^6$	0.4/0.3	-	-	-	3.9	1.7	0.43x	implicit
Figure 7	5k	0.001	0.0025	0.01	77	10^5	0.3	-	-	1.0	1.8	-	-	implicit
Figure 8	110k	0.002	0.00125	0.01	77	10^5	0.3	31.9	-	-	5.5	-	-	implicit
Figure 9	160k	0.005	0.002	0.008	2000	10^6	0.3	-	-	-	8.8	3.4	0.38x	implicit
Figure 12	200k	0.001	0.006	0.024	500/1000	10^5	0.4	-	-	-	12.3	5.6	0.46x	implicit

number of 2153851, 2153863, 2023780, 2301040, 2008915, 2244651, and 2008564.

REFERENCES

Jan Bender, Dan Koschier, Tassilo Kugelstadt, and Marcel Weiler. 2018. Turbulent micropolar SPH fluids with foam. *IEEE transactions on visualization and computer graphics* 25, 6 (2018), 2284–2295.

Jan Bender, Matthias Müller, and Miles Macklin. 2015. Position-Based Simulation Methods in Computer Graphics.. In *Eurographics (tutorials)*. 8.

Jan Bender, Matthias Müller, and Miles Macklin. 2017. A survey on position based dynamics, 2017. *Proceedings of the European Association for Computer Graphics: Tutorials* (2017), 1–31.

Jan Bender, Matthias Müller, Miguel A Otaduy, Matthias Teschner, and Miles Macklin. 2014. A survey on position-based simulation methods in computer graphics. In *Computer graphics forum*, Vol. 33. Wiley Online Library, 228–251.

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: Fusing constraint projections for fast simulation. *ACM transactions on graphics (TOG)* 33, 4 (2014), 1–11.

Jeremiah U Brackbill and Hans M Ruppel. 1986. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational physics* 65, 2 (1986), 314–343.

Oleksiy Busaryev, Tamal K Dey, Huamin Wang, and Zhong Ren. 2012. Animating bubble interactions in a liquid foam. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–8.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26 (2013), 2292–2300.

Fang Da, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2015. Double bubbles sans toil and trouble: Discrete circulation-preserving vortex sheets for soap films and foams. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–9.

Fernando de Goes, Corentin Wallez, Jin Huang, Dmitry Pavlov, and Mathieu Desbrun. 2015. Power particles: an incompressible fluid solver based on power diagrams. *ACM Trans. Graph.* 34, 4 (2015), 50–1.

Yitong Deng, Mengdi Wang, Xiangxin Kong, Shiying Xiong, Zangyueyang Xian, and Bo Zhu. 2022. A moving eulerian-lagrangian particle method for thin film and foam simulation. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–17.

Yu Fang, Minchen Li, Ming Gao, and Chenfanfu Jiang. 2019. Silly rubber: an implicit material point method for simulating non-equilibrated viscoelastic and elastoplastic solids. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–13.

Chuyuan Fu, Qi Guo, Theodore Gast, Chenfanfu Jiang, and Joseph Teran. 2017. A polynomial particle-in-cell method. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–12.

Theodore F Gast, Craig Schroeder, Alexey Stomakhin, Chenfanfu Jiang, and Joseph M Teran. 2015. Optimization integrator for large time steps. *IEEE transactions on visualization and computer graphics* 21, 10 (2015), 1103–1115.

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

Jeong-Mo Hong, Ho-Young Lee, Jong-Chul Yoon, and Chang-Hun Kim. 2008. Bubbles alive. *ACM Transactions on Graphics (TOG)* 27, 3 (2008), 1–4.

Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. 2018. A Moving Least Squares Material Point Method with Displacement Discontinuity and Two-Way Rigid Body Coupling. *ACM Transactions on Graphics* 37, 4 (2018), 150.

Sadashige Ishida, Peter Synak, Fumiya Narita, Toshiya Hachisuka, and Chris Wojtan. 2020. A model for soap film dynamics with evolving thickness. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 31–1.

Sadashige Ishida, Masafumi Yamamoto, Ryoichi Ando, and Toshiya Hachisuka. 2017. A hyperbolic geometric flow for evolving films and foams. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–11.

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.

Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. 2016. The material point method for simulating continuum materials. In *ACM SIGGRAPH 2016 Courses*. 1–52.

Gergely Klár, Theodore Gast, Andre Pradhana, Chuyuan Fu, Craig Schroeder, Chenfanfu Jiang, and Joseph Teran. 2016. Drucker-prager elastoplasticity for sand animation. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–12.

Lei Lan, Guanqun Ma, Yin Yang, Changxi Zheng, Minchen Li, and Chenfanfu Jiang. 2022. Penetration-free projective dynamics on the GPU. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–16.

Bruno Lévy. 2021. Partial Optimal Transport for a Constant-Volume Lagrangian Mesh with Free Boundaries. *arXiv preprint arXiv:2106.03936* (2021).

Minchen Li, Ming Gao, Timothy Langlois, Chenfanfu Jiang, and Danny M Kaufman. 2019. Decomposed Optimization Time Integrator for Large-Step Elastodynamics. *ACM Transactions on Graphics* 38, 4 (2019).

Xuan Li, Yadi Cao, Minchen Li, Yin Yang, Craig Schroeder, and Chenfanfu Jiang. 2022a. Plasticitynet: Learning to simulate metal, sand, and snow for optimization time integration. *Advances in Neural Information Processing Systems* 35 (2022), 27783–27796.

Xuan Li, Minchen Li, and Chenfanfu Jiang. 2022b. Energetically Consistent Inelasticity for Optimization Time Integration. *ACM Trans. Graph.* 41, 4, Article 52 (jul 2022), 16 pages. https://doi.org/10.1145/3528223.3530072

Xuan Li, Minchen Li, and Chenfanfu Jiang. 2022c. Energetically consistent inelasticity for optimization time integration. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–16.

Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. 2017. Quasi-newton methods for real-time simulation of hyperelastic materials. *Acm Transactions on Graphics (TOG)* 36, 3 (2017), 1–16.

Mickaël Ly, Jean Jouve, Laurence Boissieux, and Florence Bertails-Descoubes. 2020. Projective dynamics with dry frictional contact. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 57–1.

Miles Macklin, Matthias Müller, and Nuttapong Chentanez. 2016. XPBD: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*. 49–54.

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118.

Ken Museth, Jeff Lait, John Johanson, Jeff Budsberg, Ron Henderson, Mihai Alden, Peter Cucka, David Hill, and Andrew Pearce. 2013. OpenVDB: an open-source data structure and toolkit for high-resolution volumes. In *Acm siggraph 2013 courses*. 1–1.

Rahul Narain, Matthew Overby, and George E Brown. 2016. ADMM $\supseteq$ projective dynamics: fast simulation of general constitutive models.. In *Symposium on Computer Animation*, Vol. 1. 2016.

Matthew Overby, George E Brown, Jie Li, and Rahul Narain. 2017. ADMM $\supseteq$ projective dynamics: Fast simulation of hyperelastic models with dynamic constraints. *IEEE Transactions on Visualization and Computer Graphics* 23, 10 (2017), 2222–2234.

Ziyin Qu, Minchen Li, Fernando de Goes, and Chenfanfu Jiang. 2022. The power particle-in-cell method. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–13.

Daniel Ram, Theodore Gast, Chenfanfu Jiang, Craig Schroeder, Alexey Stomakhin, Joseph Teran, and Pirouz Kavehpour. 2015. A material point method for viscoelastic fluids, foams and sponges. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 157–163.

Juan C Simo and Thomas JR Hughes. 2006. *Computational inelasticity*. Vol. 7. Springer Science & Business Media.

Peter S. Stewart and Sascha Hilgenfeldt. 2023. Gas-Liquid Foam Dynamics: From Structural Elements to Continuum Descriptions. *Annual Review of Fluid Mechanics* 55, 1 (2023), 323–350. https://doi.org/10.1146/annurev-fluid-032822-125417

Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A material point method for snow simulation. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–10.

Deborah Sulsky, Shi-Jian Zhou, and Howard L Schreyer. 1995. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications* 87, 1-2 (1995), 236–252.

Xinlei Wang, Minchen Li, Yu Fang, Xinxin Zhang, Ming Gao, Min Tang, Danny M Kaufman, and Chenfanfu Jiang. 2020. Hierarchical optimization time integration for cfl-rate mpm stepping. *ACM Transactions on Graphics (TOG)* 39, 3 (2020), 1–16.

Joshuah Wolper, Yunuo Chen, Minchen Li, Yu Fang, Ziyin Qu, Jiecong Lu, Meggie Cheng, and Chenfanfu Jiang. 2020. AnisoMPM: Animating Anisotropic Damage Mechanics. *ACM Trans. Graph.* 39, 4, Article 37 (2020).

Joshuah Wolper, Yu Fang, Minchen Li, Jiecong Lu, Ming Gao, and Chenfanfu Jiang. 2019. CD-MPM: continuum damage material point methods for dynamic fracture animation. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.

Joel Wretborn, Sean Flynn, and Alexey Stomakhin. 2022. Guided bubbles and wet foam for realistic whitewater simulation. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–16.

Tianyi Xie, Minchen Li, Yin Yang, and Chenfanfu Jiang. 2023. A Contact Proxy Splitting Method for Lagrangian Solid-Fluid Coupling. *ACM Trans. Graph. (SIGGRAPH)* 42, 4 (2023).

Yonghao Yue, Breannan Smith, Christopher Batty, Changxi Zheng, and Eitan Grinspun. 2015. Continuum foam: A material point method for shear-dependent flows. *ACM Transactions on Graphics (TOG)* 34, 5 (2015), 1–20.

Bo Zhu, Ed Quigley, Matthew Cong, Justin Solomon, and Ronald Fedkiw. 2014. Codimensional surface tension flow on simplicial complexes. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–11.

Yongning Zhu and Robert Bridson. 2005. Animating sand as a fluid. *ACM Transactions on Graphics (TOG)* 24, 3 (2005), 965–972.